PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 :
H04L 12/43, 12/52, H04Q 11/04, H04L 12/56

A1

(11) International Publication Number: WO 00/31925

(43) International Publication Date: 2 June 2000 (02.06.00)

(21) International Application Number: PCT/SE99/02169

(22) International Filing Date: 23 November 1999 (23.11.99)

(30) Priority Data:
9804023-1 24 November 1998 (24.11.98) SE

(71) Applicant *(for all designated States except US)*: NET INSIGHT AB [SE/SE]; P.O. Box 42093, S-126 14 Stockholm (SE).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: DANIELSON, Magnus [SE/SE]; Kyrkvägen 3 A, S-182 74 Stocksund (SE). HOLMLUND, Mattias [SE/SE]; Robert Almströmsgatan 6, S-113 36 Stockholm (SE). TESSIER, Stéphane [FR/SE]; Professorslingan 11, S-104 05 Stockholm (SE).

(74) Agent: AWAPATENT AB; P.O. Box 45086, S-104 30 Stockholm (SE).

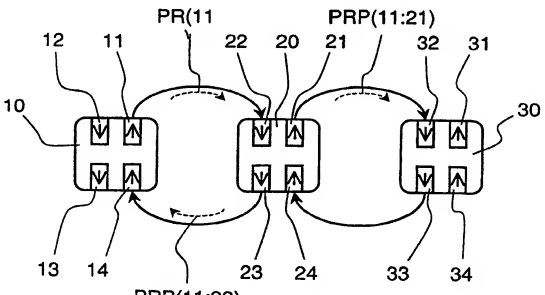
(81) Designated States: AE, AL, AM, AT, AT (Utility model), AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), DM, EE, EE (Utility model), ES, FI, FI (Utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (Utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

**Published**

*With international search report.*

*Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.*

(54) Title: METHODS AND SYSTEMS FOR DETERMINING NETWORK TOPOLOGY




(57) Abstract

The present invention relates to a method and a system for determining the topology of a network of nodes that are interconnected via unidirectional connections. According to the invention, the existence of a network loop within said network is determined using message forwarding among said nodes, and, as a result, information related to the existence of said network loop is distributed to nodes within said network.

*FOR THE PURPOSES OF INFORMATION ONLY*

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| AL | Albania | ES | Spain | LS | Lesotho | SI | Slovenia |
| AM | Armenia | FI | Finland | LT | Lithuania | SK | Slovakia |
| AT | Austria | FR | France | LU | Luxembourg | SN | Senegal |
| AU | Australia | GA | Gabon | LV | Latvia | SZ | Swaziland |
| AZ | Azerbaijan | GB | United Kingdom | MC | Monaco | TD | Chad |
| BA | Bosnia and Herzegovina | GE | Georgia | MD | Republic of Moldova | TG | Togo |
| BB | Barbados | GH | Ghana | MG | Madagascar | TJ | Tajikistan |
| BE | Belgium | GN | Guinea | MK | The former Yugoslav Republic of Macedonia | TM | Turkmenistan |
| BF | Burkina Faso | GR | Greece | | | TR | Turkey |
| BG | Bulgaria | HU | Hungary | ML | Mali | TT | Trinidad and Tobago |
| BJ | Benin | IE | Ireland | MN | Mongolia | UA | Ukraine |
| BR | Brazil | IL | Israel | MR | Mauritania | UG | Uganda |
| BY | Belarus | IS | Iceland | MW | Malawi | US | United States of America |
| CA | Canada | IT | Italy | MX | Mexico | UZ | Uzbekistan |
| CF | Central African Republic | JP | Japan | NE | Niger | VN | Viet Nam |
| CG | Congo | KE | Kenya | NL | Netherlands | YU | Yugoslavia |
| CH | Switzerland | KG | Kyrgyzstan | NO | Norway | ZW | Zimbabwe |
| CI | Côte d'Ivoire | KP | Democratic People's Republic of Korea | NZ | New Zealand | | |
| CM | Cameroon | | | PL | Poland | | |
| CN | China | KR | Republic of Korea | PT | Portugal | | |
| CU | Cuba | KZ | Kazakstan | RO | Romania | | |
| CZ | Czech Republic | LC | Saint Lucia | RU | Russian Federation | | |
| DE | Germany | LI | Liechtenstein | SD | Sudan | | |
| DK | Denmark | LK | Sri Lanka | SE | Sweden | | |
| EE | Estonia | LR | Liberia | SG | Singapore | | |

# METHODS AND SYSTEMS FOR DETERMINING NETWORK TOPOLOGY

## Technical Field of Invention

The present invention refers to methods and systems for determining a reconfigurable topology of a network of nodes having ports that are interconnected via unidirectional connections. The invention is especially applicable is the context of networks wherin several different kinds of topologies are allowed within the network.

## Background of the Invention

A communication network is a data processing system that includes a plurality of interconnected components, or nodes, such as work stations, phones, data storage devices, printers, servers, switches, routers, hubs, etc. The ports of the nodes are interconnected via connections and communicate by transmitting and receiving messages to or from ports on other nodes on such connections.

In order for a node to know how a selected destination in the network is reached, for example in which direction to send a message or set up a channel destined for an intended receiver, there is a need for the nodes to know the topology, sometimes also referred to as architecture or configuration, of the network, at least of a local portion thereof. Such information can either be provided to one or more nodes of the network using manual configuration, or it can be provided using different kinds of automized ways of having the nodes of the network discovering the network topology on thier own.

One way of providing each node with information on the network topology is to use a centralized scheme in which a central source will provide a map of the network to all other nodes of the network, for example as described in U.S. Pat. No. 5,654,958 (Natarajan). A disadvantage of this solution is that each change in the network topology has to be brought to the attention of the central source and has to be addressed by the central source

if said change is to come to the other nodes attention, automatically adding signaling overhead between the central source and the network nodes. Also, if the central source is down, updating of network topology is temporarily rendered impossible.

Another way of providing each node with information about the network topology is to use a distributed scheme in which messages, containing information pertaining to the local network topology, are exchanged between the nodes of the network. Based upon received topology messages, each node will generate and maintain its own map of the network, or at least of a local portion thereof.

U.S. Pat. No. 5,682,479 (Newhall et al.) describes an example of such a solution, wherein each network node is arranged to transmit vector-routed packets cross the network in various specified direction, each packet gathering information about the network topology along its way. The packets are then returned to the originating node with the gathered information.

U.S. Pat. No. US 5,506,838 (Flanagan) describes a similar solution wherein so-called discovery packets are forwarded from link to link within the network, thereby informing the network nodes on network topology.

As another example, U.S. Pat. No. 5,732,086 (Son-Chyay et al.) describes a method for determining a reconfigurable topology of a network by each node exchanging messages with its neighbors.

As another example, UK Patent Application GB 2,133,952 describes a method for verifying the topology of a network of nodes that are connected in multiple ring link topologies.

A disadvantage with the above-mentioned prior art schemes is that they reliy on the limitation that nodes in each case are interconnected using a predifined type of links, for example requiering that the nodes of the netowrk are interconnected via bidirectional point-to-point connections only or requiering that the nodes of

the network are connected via ring links only. Such limitations has the advantage of simplifying the design of each scheme, but also has the negative effect of limiting the applicability of the schemes.

In networks in which several different kinds of link types may exist, the task of automatically determining network topology becomes more difficult. For example in a so-called DTM (Dynamic synchronous Transfer Mode) network, ports are connected via a unidirectional connections (typically an optical fiber) to form point-to-point links (two unidirectional point-to-point connections), single ring links (each formed by one unidirectional ring link), dual ring links (each formed by two unidirectional ring links), or dual bus links (each formed by two unidirectional bus links), the latter three link types being multi-access shared links.

An object of the invention is therefore to provide a simple distributed scheme for determining network topology which allows several kinds of link types to exist in the network and that does not rely solely on bidirectional point-to-point connectivity.

Yet another object is to provide a scheme wherein the amount of messages transmitted within the network in order to determine the network topology is kept low.

Summary of the Invention

The above-mentioned objects are achieved by the invention as expressed in methods and systems according to the accompanying claims.

According to the invention, the existence of a loop within a network of nodes having ports that are interconnected via unidirectional connections is determined using message forwarding. Information gained by having determined the existence of said loop is then distributed to nodes of the network.

The invention is thus based upon the idea of regarding the network in terms of network loops (at least when

determining network topology), and to determine topology, verify topology, and distribute information related to based upon the existence of such loops.

According to a preferred embodiment, a node of the network will transmit a message, sometimes referred to herein as a topology discovery message or probe message, from an output port and will subsequently determine reception of a forwarded or reply version of said message at an input port, thereby indicating the existence of a network loop. Furthermore, each node of the network that receives a topology discovery message is preferably arranged to forward or reply to said message on at least one, typically all, of its output ports.

Preferably, each or at least a plurality of the nodes of the network will be arranged to transmit and detect messages of this kind. Furthermore, preferably all nodes of the network will be arranged to forward or reply to such messages.

As is understood from the invention, in most networks incorporating distributed control functions, even those that allow several kinds of link types to exist within the same network, a unidirectional connection between a first node and a second node will generally only form part of a valid link or topology if there exists some kind of communication path from the second node back to the first node. In other words, it must be possible to in theory draw, following in the direction of unidirectional connections/links, a line forming a loop from the first node to the second node and back to the first node.

Having determined the existence of a loop, information pertaining thereto is distributed to nodes of the network, typically nodes forming at least part of said network loop, but possible also to other nodes of the network, i.e. nodes that do not form part of said loop. According to one embodiment, this will include distributing information as to which nodes, and which ports

thereof, that form part of said network loop, wherein generation and distribution of such information is preferrably performed using message forwarding. According to another embodiment, it will include informing a neighbor node on the existens of a valid connection thereto.

Ideally, transmission and forwarding/reply using one single and comparatively small message will be enough to determine the existence of a network loop. Advantageously, such a loop will be determined even though the network, and more specifically the loop as such, may comprise a number of point-to-point connections that lack bi-directional connectivity, which in some cases would have been impossible in prior art.

According to a preferred embodiment, a node having two or more outgoing ports, and having not yet been able to determine which one of said output ports that is part of a specific network loop, is arranged to transmit two or more respective messages from respective output ports, each message identifying the respective output port (sometimes referred to herein as "reply ports"), used for transmission thereof and thereby enabling subsequent determination of which one of said two or more output ports of said node that forms part of said network loop. For example, the sending of such two or more different messages may be initiated by the reception of a topology discovery message (probe message) or may be initiated by the sending node itself.

Expressed in terms of steps performed by nodes of the network, a specific example of this embodiment comprises the steps of: transmitting a message from an output port of a first node; receiving said message, as such or in a forwarded version, at an input port of a second node; transmitting two or more modified versions (forward version or reply version) of said message from respective two or more output ports of said second node, each modified version identifying the respective output port used for transmission thereof; receiving at least

one of said modified versions of said message at said first node, thereby identifying which one of said two or more output ports that forms part of said network loop; and transmitting a message from said first node to said second node, said message identifying the output port of said second node that forms part of said network loop.

To be noted, several ports at the second node may provide a path back to the sender of the first message. For example, if the first node and the second node are both connected to the same dual ring link, there will exist two return paths from the second node to the first node, one along the same unidirectional ring as the output port that the first message was transmitted from connection is connected to, and one in the opposite direction along the other unidirectional ring. However, if the two nodes for example form part of the same single ring link or dual bus link, only one port of the second node would provide a path back to the originating node (assuming that no paths over other links are available).

Also to be noted, the so-determined reply port does not necessarily have to be the port that the repying node would generally use for transmitting data to the originating node. It is merely selected as being one of the ports that the replying node may use for sending control messages upstreams to the originating node, especially control messages regarding the probed connection (or rather the link that the connection forms part of).

An advantage of the invention is thus that a first node may determine the existence of a valid connection to second node without knowing in advance what kind of link the connection forms part of. Similariy, the second node will gain information on the valid connection and is able to reply to the first node without knowing in advance how to reach the first node nor the exact type of the links concerned in the message exchange.

According to another embodiment of the invention, link topology messages providing information on which

nodes that are connected to a multi-acces link that a unidirectional connection forms part of are propagated from one node to another on the link, each node using the output port/connection that forms part of the link to reach a downstream node on the link and using a reply port (which typically has beed identified using the above-mentioned scheme and which may or may not be part of the actual link which the topology message pertaing to) to reach an upstream node on the link.

According to yet another embodiment, information as to which nodes, and preferably also which ports thereof, that form part of a determined network is generated using message forwarding. For example, each node forwarding or replying to a received topology discovery message, or a forwarded or reply version thereof, may correspondingly include information as to the identity of the forwarding/replying node, and typically also of the output port thereof, into said message. The loop information generated in such a manner may then be distributed to the nodes forming said loop, preferably also using message forwarding.

As used herein, an interface is generally defined by an input port and an output port of a node. When a node is connected to a multi-acces ring or bus link, it is connected to the link using the input port and the output port of the same interface. Similarly, two connections connected to the same interface is generally considered herein to form part of the same link, such as a bidirectional point-to-point link, a unidirectional bus link, or a unidirectional ring link. To be noted, a unidirectional single bus link may generally as such (i.e. if lacking any other connectivity) be viewed as forming a invalid link, since there is no possibility for a downstream node to send a message to an upstream node on the link unless other connections/links are added to form an expanded topology (for example turning the single bus link into a single ring link, a dual bus link).

Although applicable in many types of networks, the invention is especially advantageous, for reasons discussed above, in networks allowing the existence of several kind of link types, such as ring, bus, as well as point-to-point links, such as in a DTM network. More information on DTM networks are found in, for example, "The DTM Gigabit Network", Christer Bohm, Per Lindgren, Lars Ramfelt, and Peter Sjödin, Journal of High Speed Networks, 3(2):109-126, 1994, and "Multi-gigabit networking based on DTM", Lars Gauffin, Lars Håkansson, and Björn Pehrson, Computer networks and ISDN Systems, 24(2):119-139, April 1992.

Concluding, the solution of probing the network in terms of network loops, determining the existence of such loops, and distributing topology information pertaining thereto to at least one node that forms part of the loop, clearly forms an inventive idea involving an inventive step.

The above mentioned and other aspects, features and details of the invention will be more fully understood from the following description of exemplifying embodiments thereof.

Brief Description of the Drawings

Exemplifying embodiments of the invention will now be described in detail with reference to the accompanying drawings, wherein:

Figs. 1a-1c show networks based on different kinds of link types;

Fig. 2 shows a flow chart of a topology discovery process according to an embodiment of the invention;

Figs. 3-5 illustrate how messages are exchange between nodes of a network in accordence with the topology discovery process shown in Fig. 2;

Fig. 6 shows a flow chart of a topology discovery process according to another embodiment of the invention;

Figs. 7a-7c illustrate how messages are exchange between nodes of a network in accordence with the exemplifying topology discovery process shown in Fig. 6;

Fig. 8a is a block diagram of components of an exemplifying network node; and

Figs. 8b and 8c illustrates topology information stored in the memory of the node in Fig. 8a.

Detailed Description of a Preferred Embodiment

Fig. 1a-1c illustrate nodes 10, 20 and 30 being capable of operating in networks based on different kinds of link types. As illustreated, node 10 comprises two interfaces, one defined an output port 11 and an input port 12 and the other defined by an output port 13 and an input port 14. Similar interfaces and ports are found on nodes 20 and 30.

In Fig. 1a, the nodes 10, 20 and 30 are interconnected via unidirectional connections 101, 102, 103, and 104 to form a dual bus link interconnected.

Connections 101 and 102 together forms a first unidirectional bus interconnecting nodes 10, 20 and 30, node 10 acting as head end of the bus and node 30 acting as terminating end thereof. Connections 103 and 104 forms a second unidirectional bus also interconnecting nodes 10-30, node 30 in this case acting as the head end and node 10 acting as terminating end thereof. Together, the bus links formed by connections 101-104 forms a double bus link. It is also noted that node 10, connection 101 from output port 11 of node 10 to input port 22 of node 20, node 20, and connection 104 from output port 23 of node 20 to input port 14 of node 10 together may be viewed as forming a network loop, as indicated by a semi-circular, dotted arrow. Likewise, node 20, connection 102 from node 20 to node 30, node 30, and connection 103 from node 30 to node 20 together may be viewed as forming another network loop 32. The double bus link may thus be viewed as being built up by three consecutive network loops 31-

33. Likewise, said second double bus link may be said to be built up by two consecutive network loops 34 and 35.

In Fig. 1b, the nodes 10, 20 and 30 are interconnected via unidirectional connections 105, 106, and 107 to form a ring link. In this case, the ring topology may thus be viewd as forming one network loop that comprises all three nodes, as indicated by the semi-circular dotted arrow in the center of the figure.

As shown in Figs 1a and 1b, when connected to form multi-access shared links, a node is typically connected to the link using the input and output port of one single interface.

As a third example, in Fig. 1c, the nodes 10, 20 and 30 are interconnected via unidirectional to form a row of two bidirectional point-to-point links, both ports of an interface of one node being connected to both ports of an interface of another node. However, the connections forming a bidirectional point-to-point link may also be viewed as together forming small network loops, as illustrated by the dotted semi-circular arrows.

A flow chart of a topology discovery process according to an embodiment of the invention will now be described with reference to Fig. 2. As is understood, the main object of this topology discovery algorithm is to determine the existence of network loops of the kind indicated in Figs. 1a-1c and to provide information related thereto to the nodes that form part of the respective network loop.

With reference to Fig. 2, the topology discovery algorithm comprises a loop detection step S10, a master announce step S20, a split-point reduction step S30, a loop list build-up step S40, a loop list distribution step S50, and a route table computation step S60.

Each of the steps of Fig. 2 will now be described with reference to an exemplifying network illustrated in Figs. 3-5, said network comprising six nodes 10, 20, 30, 40, 50, and 60. In this example, it is for simlicity

assumed that node 10 only has one interface being defined by an output port 11 and an input port 12, and that each one of nodes 20, 30, 50, and 60 in similar has one corresponding interface, wheras node 40 is assumed to comprise two interfaces, one being defined by output port 41 and input port 42 and the other being defined by output port 43 and input port 44.

As shown in Figs. 3-5, output port 11 of node 10 is connected via a unidirectional connection to input port 22 of node 20, output port 21 of node 20 is connected via a unidirectional connection to input port 32 of node 30, output port 31 of node 30 is connected via a unidirectional connection to input port 42 of node 40, and output port 41 of node 40 is connected via a unidirectional connection to input port 12 of node 10, in all forming a first single ring link. Furthermore, output port 43 of node 40 is connected via a unidirectional connection to input port 52 of node 50, output port 51 of node 50 is connected via a unidirectional connection to input port 62 of node 60, and output port 61 of node 60 is connected via a unidirectional connection to input port 44 of node 40, in all forming a second single ring link.

An example of a loop detection step S10 of the topology discovery algorithm in Fig. 2 according to the preferred embodiment of the invention will now be described with reference to Fig. 3. During the loop detection step S10, the nodes of the network will transmit so called probe messages that are used to detect the presence of loops in the network topology. A node transmits probe messages on all output ports that, are not part of already determined loops. When a node generates probe messages, each message is provided with a unique identification identifying the probe message origin, i.e. identifying the output port that the message is transmitted from, for example the unique MAC address of the output port. The nodes of the network are in this embodiment arranged to forward received probe messages on all output

ports. When forwarding a probe message, the content of the received probe message is mapped into the transmitted message. In other words, the content of the forwarded probe message will essentially be a copy of the content of the received probe message, thus identifying output port of the node that originated the probe message. The distribution of probe messages is limited by a hop-count mechanism that limits the number of hops that a probe message is forwarded over. (For simplicity, in the illustrated example, it is assumed that the number of hops that a message is allowed to travel is set to four.)

When a node receives one of its own probe message from another node, it will determine that a loop exists, and it will know which one of its input ports and output ports that forms part of this new loop. The node then becomes a so-called build-up master for the new loop and continues to the master announce step for the new loop.

In the example shown in Fig. 3, node 10 transmits a probe message PR(11) on its output port 11 to node 20, said probe message identifying the origin of the probe message. Node 20 then forwards the probe message on output port 21 to node 30, having incremented the hop-count indicated in the probe message by one. Node 30 forwards the probe message on output port 31 to node 40. Since node 40 has two output ports, it forwards the probe message on output port 41 to node 10 as well as on output port 43 to node 50. Node 50 will then forward the probe message on output port 51 to node 60. Since the maximum number of allowed hops has been reached, node 60 will decide not to forward the probe message. However, at the same time, node 10 will have received its own probe message from node 40 on input port 12 and will therefore determine that a network loop exists and that output port 11 and input port 12 are part of this new loop. Node 10 will then take on the role as build-up master and continue to the master announce step.

In the master announce step S20 of the topology discovery algorithm, when a node has determined the presence of a new loop using probe messages as described with reference to Fig. 3, it will take on the role as build-up master for the new loop. In order to let other nodes learn about the existence of the loop, the build-up master will send out a so-called master announce message (not shown) on the output port previously identified as being part of the new loop. The master announce message is forwarded by the same rules as the probe messages and serves two purposes. The first purpose is to inform other nodes about the existence of a new loop and to assign an identifier to the new loop, typically being the MAC address as mentioned above. This loop identifier is contained in all subsequent messages concerning the new loop and allows several loops to be discovered simultaneously without risking mix up of messages referring to different new loops. Upon receiving the master announce message, the other nodes become so called build-up slaves for the new loop and automatically know which of its input ports that are part of the loop. The second purpose of the master announce message is that it provides a mechanism for build-up master arbitration. If two nodes simultaneously receive their own probe messages, they will both try to take on the role as build-up master. In this preferred embodiment, this is resolved by a precedence mechanism based on the MAC addresses of the build-up masters. If a build-up master receives a master announce message (on the input port for which it is currently trying to become build-up master) from a node with a higher MAC address, it retreats, at least temporarily, and becomes build-up slave instead. If a build-up master receives a master announce message from a node with a lower MAC address, the master announce message is not forwarded.

When the build-up master eventually receives its own master announce message, it knows that all other nodes in

the new loop have become build-up slaves and that it is the only build-up master for the new loop. The build-up master then continues to the split-point reduction step.

An example of the split-point reduction step S30 of the topology discovery algorithm according to the preferred embodiment of the invention will now be described with reference to Figs. 4a and 4b. A so-called split-point node is a node that has two or more connected output ports, in this case node 40. When a node forwards a probe message or a master announce message, it must forward the messages on all its output ports, since it does not know which port that is part of the new loop. The goal of the split-point reduction step is to determine which one of the output ports of the split-point node that forms part of the new loop.

The split-point reduction step is started by the build-up master, which will send out a so called split-point announce message on the output port where it previously sent out the probe and master announce messages. The split-point announce message is provided with an identifier of the output port that it was sent on.

Split-point announce messages are forwarded according to the following rule: If the forwarding node has only one single output port, or if it already knows which output port that is part of the new loop (i.e. it has already been "resolved" as discussed below), the split-point announce message is forwarded in an unmodified version via the correct (or only) output port. Otherwise, the node is a considered a split-point node. It then sends out its own new split-point announce messages on all its output ports, instead of the received split-point announce message. Moreover, each new split-point announce message contains an identifier of the output port that it was sent on.

When the build-up master receives a split-point announce message with an identifier for an output port of another node, it knows that that output port is part of

the new loop. The build-up master informs the split-point node about this by transmitting a so-called split-point reduce message identifying that output port. Split-point reduce messages are forwarded in the same way as probe and master announce messages. When a split-point node receives a split-point reduce message identifying one of its output ports, it knows that that port is part of the new loop. The split-point has now been resolved, and the split-point reduce message need not be forwarded. Consequently, when the next split-point announce message is received at the node, it is forwarded unmodified on the now determined output port for the new loop.

Moreover, when a split-point node has been informed about which of its output ports that is part of the new loop, it sends out a so called release branch message on all other output ports. This is done to inform build-up slaves downstream from those ports that they are not part of the loop and they can now remove all protocol state regarding the loop and try to establish other loops instead.

Directly after sending out a split-point reduce message, the build-up master sends out a new split-point announce message to find the next split-point node. The split-point reduction step thus continues by reducing one split-point at a time starting from the split-point closest to the input port of the build-up master and working its way back to the output port of the build-up master.

When the build-up master finally receives one of its own split-point announce messages, it knows that there are no more split-points in the loop. All the nodes forming part of the loop now knows which of its ports that are part of the loop, and all messages regarding the new loop is therefore transmitted only to the nodes involved in the loop. So far, however, each node only has information about its own ports, i.e. no node has complete knowledge of all nodes in the new loop (except in

very simple topologies). The build-up master therefore continues to the so-called loop list buildup step.

In the example shown in Figs. 5a and 5b, the build-up master node 10 starts by sending out a split-point announce message SPA(11) on the output port 11 to node 20. The split-point announce message SPA(11) is provided with an identifier of the output port 11 that it was sent on. The message SPA(11) is forwarded by nodes 20 and 30 to node 40. However, as node 40 has two connected output ports, node 40 sends out its own new split-point announce messages SPA(41) and SPA(43) on its respective output ports, each message identifying the output port that it was sent on. When the build-up master node 10 receives the message SPA(43) from node 40 in Fig. 5a, it knows that the identified output port 41 is part of the new loop. As shown in Fig. 5b, the master node 10 therefore informs the split-point node 40 about this by transmitting a split-point reduce message SPR(41) identifying the output port 41, said message being transmitted/forwarded the same way as the previous messages. When node 40 receives the split-point reduce message SPR(41) identifying its output port 41, it knows that port 41 is part of the new loop. Node 40 then sends out a release branch message RB on the remaining output port 43. As node 50 and 60 then receive the release branch message RB, they cease to be build-up slaves and may start searching for other loops. Directly after sending the split-point reduce message SPR(41), the build-up master 10 sends out a new split-point announce message (not shown). As the split point at node 40 has now been resolved, the new split point announce message will be forwarded all the way back to the master node 10. The master node 10 will therefore determine that there are no more split-points in the loop.

An example of the loop list build-up step of the topology discovery algorithm according to the preferred embodiment of the invention will now be described with

reference to Fig. 5. During the loop list build-up phase, the build-up master collects information about which nodes, and ports thereof, that are part of the new loop using message forwarding. The build-up master initiates the loop list build-up step by sending out an empty loop list (LB in Fig. 6) on the output port for the new loop. Each build-up slave on the loop path to the build-up master's input port adds itself to the loop list and forwards the new loop list to the next node. When the loop list reaches the build-up master again, the build-up master adds itself to the end of the loop list. The loop list build-up phase is now finished and the build-up master has complete knowledge of the topology of the loop. It can then move on to the loop list distribution step.

During the loop list distributing step, the build-up master informs all the nodes forming part of the new loop about the topology of the loop using message forwarding. The same message format is used to distribute the list as was used during the loop list build-up phase. The build-up master transmits the list LD on the output port where the new loop has been established. Each build-up slave node that receives a loop list under distribution must forward the list to the output port belonging to the same loop that the list arrived on. The build-up master that has originated the loop list must make sure that the list comes back via the input port belonging to the same loop as the output port that the list was originated on. If the loop list does not arrive within a configured time interval, or if an error is detected by a node during the loop list distribution, the list is re-originated.

When the build-up master has received the full and correct loop list as transmitted, it will cease to operate as build-up master, consider the new loop to be up and the loop state fully built. Correspondingly, when a build-up slave has received the full and correct loop list, it will cease to operate as a build-up slave, consider the loop to be up and the loop state fully built.

The (newly ceased) build-up master and slaves then continue to the routing table computation step.

In the routing table computation step, when a node has received and accepted a new loop list, it will use it to compute an updated route table based on the available loops considered valid. The route table will contain one item for each reachable node. Each item typically contains the output port that must be used to reach the destination and the MAC address of the input interface of the destination.

For example, according to an alternative embodiment, the loop detection step, the split-point-reduction step, and the loop list build-up step is integrated into one single step, wherein the rules for handling a loop detection message will include the split-point and loop list build-up features. An advantage of such a scheme is that it limits the amount of messages transmitted between the nodes of the network. On the other hand, it increases message size and processing.

A flow chart of a topology discovery process according to another embodiment of the invention will now be described with reference to Fig. 6. As is understood, the main object of this topology discovery algorithm is to determine the existence of valid connections/link by detecting network loops of the kind indicated in Figs. 1a-1c and to provide information related thereto to one or more nodes that form part of the respective link.

With reference to Fig. 6, the topology discovery algorithm comprises a loop detection step S110, a link announce step S120, a link list distribution step S130, and a route table computation step S140. The process will now be described in detali with reference to an exemplifying network illustrated in Figs. 7a-7c, said network comprising three nodes 10, 20, and 30, interconnected to form a dual bus link, i.e. in similar to the network described above with reference to Fig. 1a. The description will in this case exemplify topology determining

actions taken by the nodes in the network, each action being discribed in relation to the event that causes the action. Typically, the actions described below will be decided upon by one or more control functions in each node.

In the loop detection step S110 of Fig. 6, each node will regularly transmit so-called probe messages on all output ports for which no valid connections exists. Each probe message will be provided with a Link Identifier identifying the output port that the message is transmitted from, for example using the unique MAC address of the port, thus indirectly identifying the link that the message is transmitted on. This is illustrated by the probe message PR(11) in Fig. 7a, including an identification of the output port 11 that it is transmitted from. The purpose of a probe message is to find out if a valid connection has been established for the output port that the probe message is transmitted from. A probe message is in this embodiment only sent to neighbor nodes and is not as such forwarded to reach other nodes. The reason for this is that, in this embodiment, the topology discovery is based upon having each node discover its downstream neighbor. However, the scheme could just as well be modified to let each node discover nodes further downstream, by for example allowing the probe message as such to be forwarded one or more hops.

When a node receives a probe message, it will reply thereto by transmitting probe replies on all it's output ports. For each probe reply, it will include the Link Identifier of the probe message as well as a Reply Port Identifier identifying the output port that the probe reply message is transmitted from. This is illustrated in Fig. 7a by the two replies PRR(11:21) and PRR(11:23) that node 20 tranmits as a result of having received the probe message PR(11), each reply including the identification (11) identifying the probe message as well as an identification (21 and 23) of the respective output ports 21

and 23 on which the reply is sent. For the probe reply that is transmitted from the output port that forms part of the same interface as the input port at which the probe message was received, include a flag identifying the reply as being sent in so-called "bypass mode". The reason for including this flag will be described below. The reason for transmitting probe replies on all output ports is that the node has no way of telling in advance which one or more the output ports that provides a path back to the node that sent the probe message since several different topologies are allowed in the network.

When a node receives a probe reply, it will first of all determine, using the Link Identifier included in the probe reply message, whether or not the probe reply is a reply to a probe message that the node itself has was the sender of, i.e. if the Link Idenifier included in the probe reply message identifies an output port of the node. If the answere is no, the node will forward the probe reply on the output port of the same interface as it was received on. The reason for not forwarding the probe reply on all output ports in this embodiment, is that if the node itself wasn't the intended recipient, the path back to the intended recipient, given the allowed topologies, should be along the same unidirectional link that the message was received on. However, an embodiment wherein the probe reply is forwarded on all output ports could also be used, but then some kind of mechanism would preferably be addedd to avoid messages from being forwarded forever within the network.

If however the node determines, using the Link Identifier included in the probe reply, that the received probe reply is a reply to a probe message transmitted from an output port of the node, it will initiate the link announce step S120 of Fig. 6 by transmittin a link detected message, including the Link Identifier and the Reply Port Identifier of the probe reply (in this case port 23), from the output port identified by the Link

Identifier, i.e. the port that the probe message was originally transmitted from. This is illustrated in Fig. 7b by the originating node 10 sending a link detection message LD(11:23) on the same port as it previously sent the probe message PR(11) on. Also, if the answere is yes and the probe reply includes a flag identifying the reply as being sent in so-called "bypass mode", the node will include, in the link detected message, a flag identifying that the receiver shall not originate a link toplogy message, the reason for which being described below.

When a node receives a link detected message it will: a) conclude that a valid connection/link exists from an upstream node to the input port at which the link detected message is received; b) conclude that the upstream neighbor node identifies its ouput port for this link using the Link Identifier included in the link detected message; c) determine, using the Reply Port Identifier included in the link detected message, which output port, referred to below as reply port, that shall be used when sending control messages to the upstream neighbor node regarding the the connection/link; e) transmit a link detected acknowledgement message from the reply port to reach the upstream neighbor node, said link detected acknowledgement message including the Link Identifier, as illustrated by the message ACK(11) in Fig. 7b. Furthermore, if the link detected message does not include a flag identifying that the receiver shall not originate a link toplogy message, it will transmit (originate) a link topology message from the reply port to the upstream neigbor node on the link, said link topology message including i) the Link Identifier, ii) the stored list of nodes that, as far as the node is aware, are connected to the link identified by the Link Identifier, and iii) a flag designating the list to be an upstream distributed list. This is illustrated in Fig. 7c by the link topology messgae LT(11:20,30) transmitted by node 20 to node 10 using the determined reply port 23 and illu-

strates the link distribution step S130 of Fig. 6. Consequently, the reception of the link detected message informs a node that there now exists a link, comprising one or more node, upstreams from the input port on which the message is received. To provide the upstreams nodes with any existing information that the node has regarding itself and possible other downstream nodes, it sends the link topology message in the upstream direction using the reply port. For example, if the link segment 20-30 between node 20 and 30 existed prior to the establishment of the connection between node 10 and 20, node will, after receiving the link detected message regarding the new connection 10-20 send the node list 20-30 upstreams, thereby informating upstream nodes on the known topology downstream of the new connection.

Similarily, when the original sender of the probem essage receives the link detected acknowledgement message sent out from the reply port of the downstream node, it will conclude that a valid connection exists from the output port identified by the Link Identifier included in the link detected acknowledgement message. To inform downstream nodes now accessable via the new connection on the known topology upstream of the new connection, it will transmit (originate) a link topology message from said output port to the downstream neigbor node on the link, said link topology message including i) the Link Identifier, ii) the stored list of nodes that, as far as the originating node is aware, are connected to the link identified by the Link Identifier, and iii) a flag designating the list to be a downstream distributed list. This is illustrated in Fig. 7b by the message LT(11:10) that is transmitted from node 10 to node 20 Consequently, returning to the example above, if the link segment 20-30 existed prior to the establishment of the connection 10-20, node 10 will, after receiving the link detected message regarding the new connection 10-20 send the node

list 10 downstreams, thereby informing downstream nodes on the known topology upstream of the new connection.

The purpose of the link topology messages is that they shall be forwarded in the downstream/upstream direction, to be used to update each node on the link on the new topology. To be noted, if the newly established/-detected connection closes a single ring link, there is no need to send link topology messages both downstream along the link and upstream using the reply ports, as the downstream message will very efficiently be forwarded to all nodes on the link, and that is the reason for including the above mentioned flag for "bypass" (in the probe reply) and the corresponding flag in the link detected message, thereby instructing the receiver of the link detected message not to transmit (originate), as described above, any link topology message upstreams.

Consequently, when receiving a link topology message, a node will determine whether or not it provides new information on the topology of the link identified by the Link Identifier included in the message as compared to topology information already stored at the node. If the answere is yes, the node will update the stored list designating nodes connected to the link accordingly. Also, if the answere is yes and the flag provided in the link topology message defines it as being a downstream distributed list, it will transmit a similar link topology message on the output port belonging to the same interface as the input port on which the link distributed message was received, and it will include therein i) the Link Identifier identifying said output port, ii) the updated list of nodes, and iii) a flag designating the list to be a downstream distributed list. Returning to the example mentioned above, when node 20 receives the downstream topology message from node 10 identifying the topology (10) known to node 10, node C will be updated on that the entire known link now comprises nodes 10, 20 and 30 (the existence of node 30 was assumed already known to

node 20), and will transmit a link topology message with this topology information (10-20-30) to its downstream neighbor node 30, as is illustrated by the messag LT (21:10,20,30).

Alternatively, if the received topology information is new and the flag provided in the link topology message defines it as being an upstream distributed list, the node will transmit a similar link topology message on the reply port to the upstream neighbor node and include therein i) the Link Identifier used by the upstream neighbor node to identify the port/link to the originating node, ii) the updated list of nodes, and iii) a flag designating the list to be an upstream distributed list.

However, if a received link topology message does not provide new information as compared to topology information already stored at the node, it will not transmit any correspoingin link topology message, thereby stopping the new link topology information/message from possibly circulating forever in a ring/loop.

To be noted, in alternative embodiment, the topology information distributed in downstream/upstream topology messages could, as an alternative or addition to information explicitly identifyhing nodes, include information identifying interfaces, ports, or the like, identifying the topology of the link. It could also include information on the topology of other links that the one primarily addressed.

Moreover, each node will regularly transmit verify messages (not shown) on all output ports for which valid connections exists. Each probe message verify message will be provided with a Link Identifier identifying the output port that the message is transmitted from. The purpose of a verify message is to verify that a connection that has already been determined valid still exists from the port that the verify message is transmitted from. When receiving a verify message, a node will transmit a verify acknowledgement message on the reply port

associated with the Link Identifier included in the verify message. The upstream node will expect to receive the verify acknowledgement message within a certain period of time. If the message isn't received wihtin this period of time, it will determine the connection to no longer be valid, and will therefore originate a new link topology message in the upstream direction (using the reply port to its upstream neighbor node) to inform upstream nodes on the link at issue on the missing connection. Similarily, each node having an upstream neighbor will expect to receive a verify message regularly. If a new verify message isn't received within a certain period of time, it will determine the connection to the upstream node to no longer be valid, and will therefore originate a new link topology message in the downstream direction of the link at issue to inform downstream nodes on the missing connection.

Fig. 7 is a block diagram exemplifying general components of a node used in the networks discussed above. The node comprises a first interface defined by output port 111 and input port 112, as well as a second interface defined by output port 113 and input port 114. The two interfaces are connected to a switch core 115 that provides switching of data between the interfaces as well as to/from the interfaces and a control processor 116 of the node. Furthermore, each intefaces as such will typically also be provided with means for bypassing/-switching data from its own input port to its own output port. The control processor typically provides the above mentioned control function that handles tranmitted and received messages of the kind described above, and uses information in the message to update a memory 117 that stores topology information. Note, however, that such a control function and memory need not be centralized within the node, handling topology discovery operation with respect to all interfaces of the node. The control fucntion and/or memory storage could just as be implemen-

ted a several parallel control functions, each for example operating in relateion to a respective interface of the node.

Finally, Figs. 8b and 8c illustrates entries of the kind found in the memory 117 of Fig 8a, exemplified with the content as found in node 10 and node 20, respectivly, after having determined the existence of the connection that is discussed with reference to Figs. 7a-7c. As is illustrated, after the new connection from port 11 to port 22 has been detected and topology information pertaining therego has been distributed, the memory of node 10, illustrated in Fig. 8b, indicates that the link to which port 11 (link ID) is connected comprises nodes 10, 20 and 30. As node 10 itself is the most upstream node on the link, no reply port exist to any upstream node. Similarily, the memory of node 20, illustrated in Fig. 8c, indicates that the link to which its port 21 is connected comprises nodes 10, 20 and 30, and that node 20 can use its port 23 as reply port to reach its upstream neighbor node 10.

As understood by those skilled in the art, the above mentioned steps may be altered, modified, and/or integrated. Furthermore, steps may be added or excluded based upon the desired functionality within the scope of the invention, which is defined by the accompanying claims.

Based upon the inventive idea, many diffenet topology message handling rules may be used, for example deteriming how and when to send probe messages, how and when to look for new connections, and so on, the scope of the invention of couse not being limited to the specific embodiment described in detail above.

Hence, the decision regarding how to actually realize and implement the invention will typically depend upon how the explicit network type will be positively or negatively affected by aspects such as the amount of messages transmitted within the network, message size,

the amount of message processing, the changing and/or maintaining of states at each node, and so on.

CLAIMS

1. A method for determining the topology of a network of nodes that are interconnected via unidirectional connections, said method comprising the steps of:

determining the existence of a network loop within said network using message forwarding among said nodes; and

distributing information related to the existence of said network loop within said network.

2. A method as claimed in claim 1, wherein said step of determining the existence of a network loop comprises the steps of:

transmitting a message from an output port of a first node; and

receiving a forwarded/reply version of said message at an input port of said first node.

3. A method as claimed in claims 1 or 2, wherein said step of determining the existence of a network loop comprises the steps of receiving a message at an input port of a node and forwarding said message on one or more output ports thereof.

4. A method as claimed in any one of the preceding claims, wherein said step of distributing information on the existence of said network loop comprises using message forwarding for distributing said information.

5. A method as claimed in any one of the preceding claims, wherein said information comprises information as to which nodes that form part of said network loop.

6. A method as claimed in any one of the preceding claims, wherein said information comprises information as to which ports that form part of said network loop.

7. A method as claimed in any one of the preceding claims, including the steps of:

transmitting two or more messages from respective two or more output ports of a node, each message identifying the respective output port used for transmission thereof; and

receiving a message referring to one of said two or more messages, thereby identifying which one of said two or more output ports of said node that forms part of said network loop.

8. A method as claimed in any one of the preceding claims, including the steps of:

transmitting a message from an output port of a first node;

receiving said message, as such or in a forwarded version, at an input port of a second node;

transmitting two or more modified versions of said message from respective two or more output ports of said second node, each modified version identifying the respective output port used for transmission thereof;

receiving one of said modified versions of said message at said first node, thereby identifying which one of said two or more output ports that forms part of said network loop; and

transmitting a message from said first node to said second node, said message identifying the output port of said second node that forms part of said network loop.

9. A method as claimed in any one of the preceding claims, wherein forwarding a message comprises the step of including information as to the identity of the forwarding node into said message.

10. A method as claimed in any one of the preceding claims, wherein forwarding a message comprises the step

of including information as to the identity of the output port that said message is transmitted from into said message.

11. A system for determining the topology of a network of nodes that are interconnected via unidirectional connections, comprising a first node that is arranged to transmit a message from an output port thereof, to determine the existence of a network loop within said network by determining reception of a forwarded/reply version of said message at an input port thereof, and, as a result, to distribute information related to the existence of said network loop to nodes within said network.

12. A system as claimed in claim 11, comprising one or more second nodes being arranged to forward said message on one or more output ports thereof when receiving said message on an input port thereof.

13. A system as claimed in claims 11 or 12, wherein said nodes are arranged to distribute said information on the existence of said network loop by message forwarding.

14. A system as claimed in any one of claims 11-13, wherein a node that has two or more outgoing ports is arranged to transmit two or more respective messages from respective output ports, each message identifying the respective output port used for transmission thereof and thereby enabling subsequent determination of which one of said two or more output ports of said node that forms part of said network loop.

15. A system as claimed in any one of claims 11-14, comprising:

a first node transmitting a first message;

a second node receiving said message, as such or in a forwarded version, and transmitting two or more

modified versions of said message from respective two or more output ports, each modified version identifying the respective output port used for transmission thereof,

wherein said first node is arranged to identify which one of said two or more output ports, of said second node, that forms part of said network loop by determining reception of one of said modified versions of said message at said first node, and, as a result, to transmit a message from said first node to said second node, said message identifying the output port of said second node that forms part of said network loop.

16. A method for determining a reconfigurable topology, at least locally, of nodes in a communication network, each node comprising one or more interfaces, each interface being defined by an input port and an output port connectable to other nodes via unidirectional connections, a node directly connected to another node via a unidirectional connection herein being referred to as a neighbor node, said method comprising the steps of:

transmitting probe messages from output ports of an originating node, each probe message being provided with information identifying the output port that it is transmitted from;

receiving probe replies at input ports of the originating node, each probe reply including information identifying a probe message that the probe reply is a reply to as well as information identifying the probe reply; and

determining if a valid connection exists from an output port of the originating node to a neighbor node by determining if a received probe reply identifies a probe message that has previously been sent from an output port of the originating node and if so causing transmission of a link detected message from said output port, said link detected message being provided with information identifying the probe reply, thereby acknowledging to the

neighbor node that said connection exists and, by the provision of said information identifying the probe reply, making it possible for said neighbor node to determine which of its output ports that it may use for sending information to the originating node regarding said connection.

17. A method as claimed in claim 16, wherein any two unidirectional connections connected to the same interface herein being considered to form part of the same link, said method further comprising the steps of:

storing, for each interface of said originating node, topology information identifying nodes that, as far as the originating node is aware, are connected to the same link as said interface; and

transmitting, if having determined that a new valid connection exists from an output port of the originating node to another node, a link topology message from said output port, said link topology message being provided with information identifying said connection as well as topology information identifying nodes that, as far as the originating node is aware, are connnected to the link that said connection forms part of.

18. A method as claimed in claim 16 or 17, further comprising the step of forwarding a received probe reply to other nodes if the probe reply is determined not to identify a probe message that has previously been sent from an output port of the originating node.

19. A method as claimed in claim 18, wherein, if a received probe reply is determined not to identify a probe message that has previously been sent from an output port of the originating node, said probe reply is forwarded only on the output port that is part of the same interface as the input port at which the probe reply was received.

20. A method as claimed in claim 16, 17, 18, or 19, wherein said probe messages are sent from the originating node to its neighbor nodes only.

21. A method as claimed in claim 16, 17, 18, 19, or 20, further comprising the steps of:

receiving probe messages at input ports of the node, each probe message including information identifying the probe message; and

transmitting, for each received probe message, probe replies on all output ports of the node, each probe reply including said information identifying the probe message as well as information identifying the output port that the probe reply is sent from.

22. A method as claimed in claim 21, wherein said step of transmitting a probe reply on all output ports for each received probe message comprises including, in the probe reply sent on the output port that is part of the same interface as the input port at which said probe message was received, information identifying that the probe reply is transmitted from the same interface as the probe message was received at.

23. A method as claimed in claim 21 or 22, further comprising the steps of:

receiving link detected messages at input ports of the node, each including information identifying an output port that the probe reply that caused the sending of the link detected message was sent from;

determining that a valid connection exists from a neighbor node to the input port that the link detected message is received at, including storing information designating that said output port identified in the link detected message may be used for sending information to the neighbor node regarding said connection.

24. A method as claimed in claim 23, wherein any two unidirectional connections connected to the same interface herein being considered to form part of the same link, said method further comprising the steps of:

storing, for each interface of said originating node, topology information identifying nodes that, as far as the originating node is aware, are connected to the same link as said interface; and

transmitting, when having determined that a valid connection exists from a neighbor node to an input port that a link detected massage is received at, a link topology message from the output port identified in said link detected message, said link topology message including information identifying which connection that the topology message pertains to as well as topology information identifying nodes that, as far as the originating node is aware, are connnected to the link that said connection forms part of.

25. A method as claimed in claim 24, wherein said step of transmitting a link topology message when having determined that a valid connection exists from a neighbor node to an input port that a link detected massage is received at is performed only if said link detected message does not indicate that no link topology message shall be sent as a result of the received link detected message.

26. A method for locally distributing topology information among nodes in a communication network, each node comprising one or more interfaces, each interface being defined by an input port and an output port connectable to other nodes via unidirectional connections, any two unidirectional connections connected to the same interface herein being considered to form part of the same link, a node directly connected to another

node via a unidirectional connection herein being referred to as a neighbor node, said method comprising the steps of:

storing, for each interface of a node, topology information identifying nodes that, as far as the node is aware, are connected to the same link as the interface as well as information identifying a suggested output port of the node to be used for sending topology information to an upstream neighbor node on said link;

receiving link topology messages at input ports of the node, each message including information identifying the link that it pertains to as well as topology information identifying nodes connnected to said link;

updating, for each received link topology message, stored topology information in accordence with topology information provided by the received link topology message; and

transmitting, for each received link topology message, a link topology message from the output port of the interface at which the received link topology message was received if the received link topology message was sent by an upstream neighbor node on the link that the received link topology message refers to, or, if the received topology message was sent by a downstream neighbor node on the said link, from said suggested output port to be used for sending topology information to the upstream neighbor node on said link, the transmitted topology message including information identifying the link that it pertains to as well as topology information identifying nodes that, as far as the node is aware, are connected to said link.

27. A method as claimed in claims 26, wherein said updating step is performed only if the topology information provided in the received topology message is new as compared to already stored topology information regarding the link that the topology message pertains to.

28. A method as claimed in claim 26 or 26, wherein said transmitting step is performed only if the received link topology message provides topology information that is new as compared to already stored topology information regarding the link that the topology message pertains to.

29. A method as claimed in claim 26, 27 or 28, wherein topology information included in the link topology message that is transmitted in said transmitting step is selected so as to reflect the accumulated topology information provided by the received link toplogy message and the topology information stored with respect to the subject link prior to the reception of the received link toplology message.

30. A system for determining a reconfigurable topology, at least locally, of nodes in a communication network, each node comprising one or more interfaces, each interface being defined by an input port and an output port connectable to other nodes via unidirectional connections, a node directly connected to another node via a unidirectional connection herein being referred to as a neighbor node, an originating node further comprising:

transmitter means for transmitting probe messages from output ports of the originating node, each probe message being provided with information identifying the output port that it is transmitted from;

receiver means for receiving probe replies at input ports of the originating node, each probe reply including information identifying a probe message that the probe reply is a reply to as well as information identifying the probe reply; and

logic means for determining if a valid connection exists from an output port of the originating node to a neighbor node by determining if a received probe reply identifies a probe message that has previously been sent

from an output port of the originating node and if so causing transmission of a link detected message from said output port, said link detected message being provided with information identifying the probe reply, thereby acknowledging to the neighbor node that said connection exists and, by the provision of said information identifying the probe reply, making it possible for said neighbor node to determine which of its output ports that it may use for sending information to the originating node regarding said connection.

31. A system as claimed in claim 30, wherein any two unidirectional connections connected to the same interface herein being considered to form part of the same link, said method further comprising:

memory means for storing, for each interface of said originating node, topology information identifying nodes that, as far as the originating node is aware, are connected to the same link as the interface,

said logic means, if having determined that a valid connection exists from an output port of the originating node to another node, being arranged to cause transmission of a link topology message from said output port, said link topology message being provided with information identifying said connection as well as topology information identifying nodes that, as far as the originating node is aware, are connnected to the link that said connection forms part of.

32. A system as claimed in claim 30 or 31, said logic means, if determining that a receiced probe reply does not identify a probe message that has previously been transmitted from an output port of the originating node, being arranged to cause forwarding of the probe reply to other nodes.

33. A system as claimed in claim 32, said logic means, if determining that a received probe reply does not identify a probe message that has previously been sent from an output port of the originating node, being arranged to cause forwarding of said probe reply only on the output port that forms part of the same interface as the input port at which the probe reply was received.

34. A system as claimed in claim 30, 31, 32, or 33, wherein said probe messages are transmitted from the originating node to its neighbor nodes only.

35. A system as claimed in claim 30, 31, 32, 33, or 34, said logic means being arranged, when a probe message has been received at an input port of the node, said probe message including information identifying the probe message, to cause transmission of a probe reply on each output port of the node, each probe reply being provided with said information identifying the probe message as well as information identifying the output port that the probe reply is transmitted from.

36. A system as claimed in claim 35, said logic means being arranged, when a link detected message has been received at an input port of the node, said message including information identifying an output port that the probe reply that caused the sending of the link detected message was sent from, to determine that a valid connection exists from a neighbor node to the input port that the link detected message was received at and to cause storing, in said memory means, of information designating that said output port identified in said link detected message may be used for sending information to the neighbor node regarding said connection.

37. A system as claimed in claim 36, wherein any two unidirectional connections connected to the same inter-

face herein being considered to form part of the same link, said logic means geing arranged, when having determined that a new valid connection exists from a neighbor node to an input port that a link detected massage was received at, to cause transmission of a link topology message from the output port identified in said link detected message, said link topology message including information identifying said connection as well as topology information identifying nodes that, as far as the originating node is aware, are connnected to the link that said connection forms part of.

38. A system for locally distributing topology information among nodes in a communication network, each node comprising one or more interfaces, each interface being defined by an input port and an output port connectable to other nodes via unidirectional connections, any two unidirectional connections connected to the same interface herein being considered to form part of the same link, a node directly connected to another node via a unidirectional connection herein being referred to as a neighbor node, a node comprising:

memory means for storing, for each interface of a node, topology information identifying nodes that, as far as the node is aware, are connected to the same link as the interface as well as information identifying a suggested output port of the node that the node may use for sending topology information to an upstream neighbor node on said link;

receiver means for receiving link topology messages at input ports of the node, each message including information identifying the link that it pertains to as well as topology information identifying nodes connnected to said link;

transmitter means for transmitting link topology messages from output ports of the node, each message including information identifying the link that it

pertains to as well as topology information identifying nodes that, as far as then node is aware, are connnected to said link;

logic means for updating topology information stored in said memory means in accordence with topology information provided by received link topology messages and for causing, for each received link topology message, transmission of a link topology message from the output port of the interface at which the received link topology message was received if the received link topology message was sent by an upstream neighbor node on the link that the received link topology message refers to, or, if the received topology message was sent by a downstream neighbor node on said link, from said suggested output port to be used for sending topology information to the upstream neighbor node on said link.

39. A system as claimed in claims 38, said logic means being arranged to update topology information stored in said memory menas only if topology information provided in received topology messages is new as compared to already stored topology information.

40. A system as claimed in claim 38 or 39, said logic means being arranged to cause said transmission of a link topology message only if the received link topology message provides topology information that is new as compared to already stored topology information regarding the link that the topology message pertains to.

41. A system as claimed in claim 38, 39, or 40, said logic means being arranged to select topology information to be included in a topology message to be transmitted from said node as a result of the reception of a link topology message so as to reflect the accumulated topology information provided by the received link toplogy message and topology information already stored with

respect to the subject link prior to the reception of the received link toplology message.

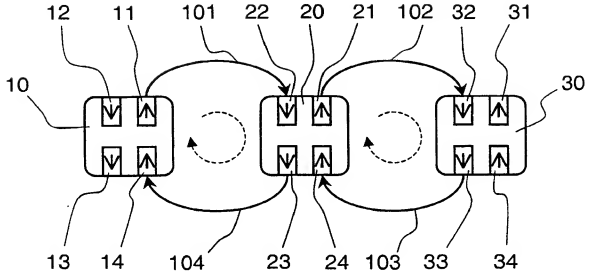

12
11
101
22
20
21
102
32
31
10
30
13
14
104
23
24
103
33
34

Fig. 1a

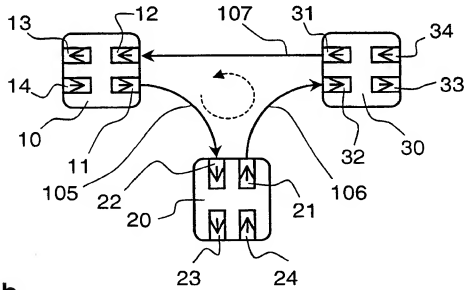

13
12
107
31
34
14
33
10
11
32
30
105
22
106
20
21
23
24

Fig. 1b

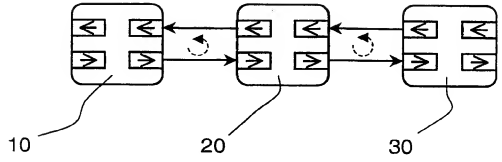

10
20
30

Fig. 1c

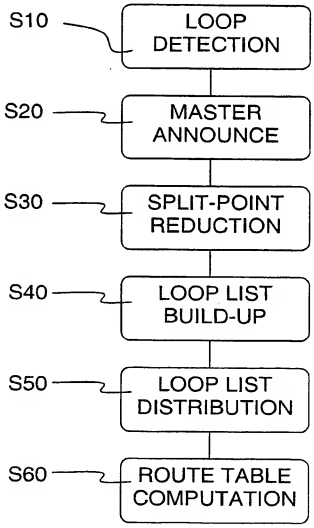

S10
LOOP DETECTION
S20
MASTER ANNOUNCE
S30
SPLIT-POINT REDUCTION
S40
LOOP LIST BUILD-UP
S50
LOOP LIST DISTRIBUTION
S60
ROUTE TABLE COMPUTATION

Fig. 2

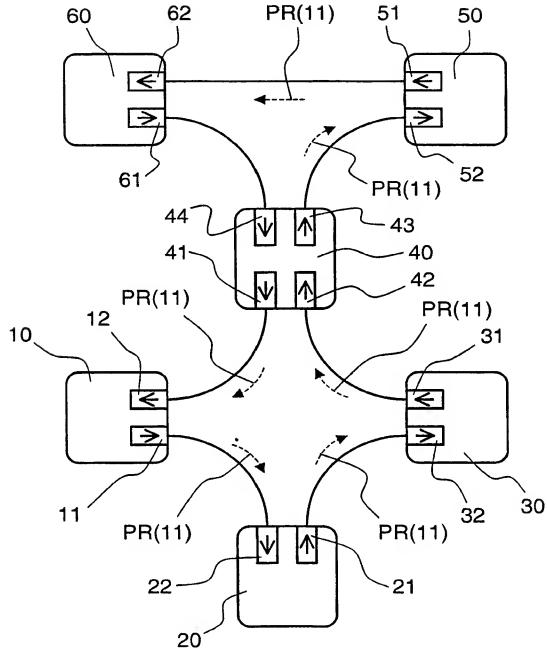



Fig. 3

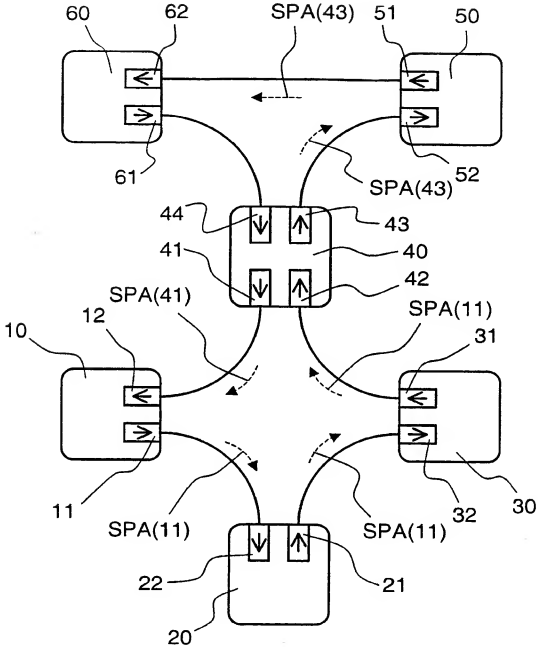



Fig. 4a

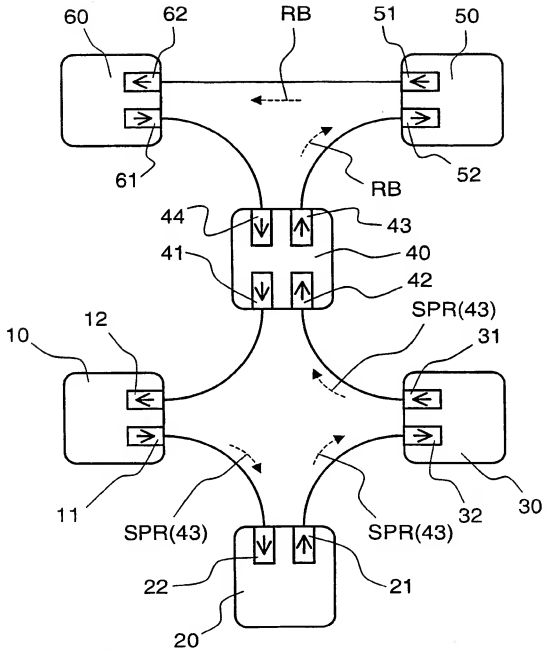



Fig. 4b

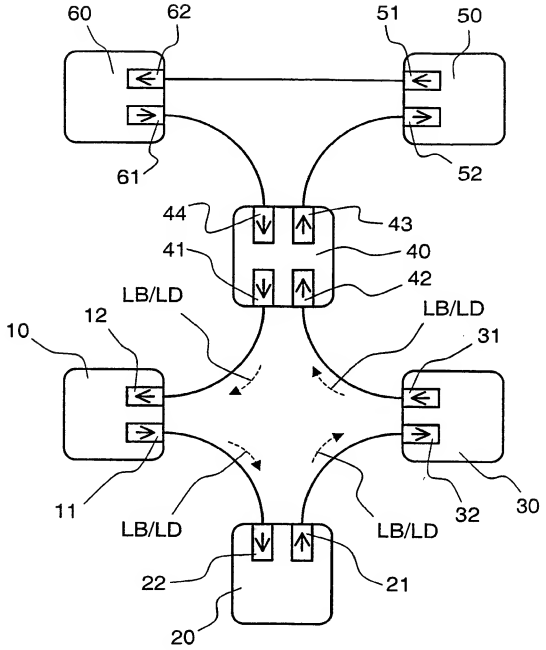



Fig. 5

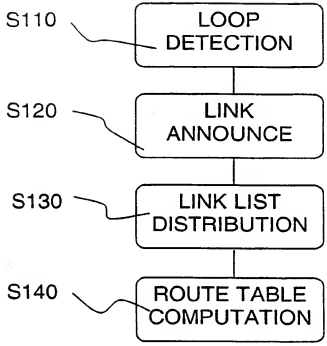



Fig. 6

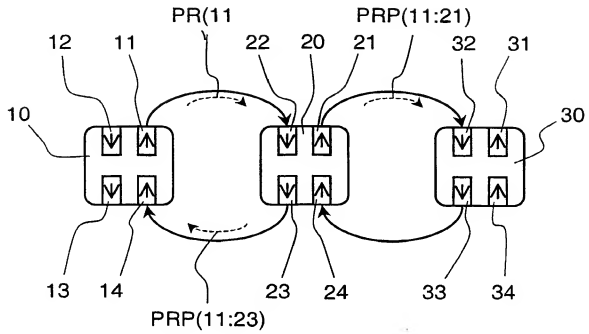



Fig. 7a

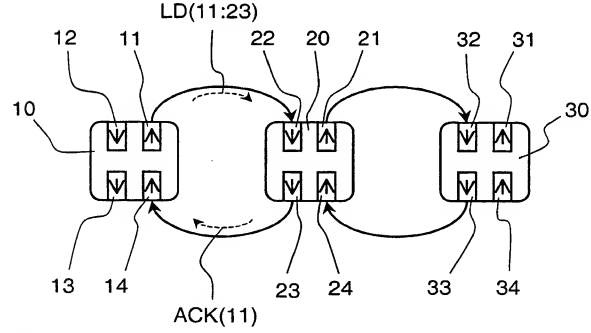



Fig. 7b

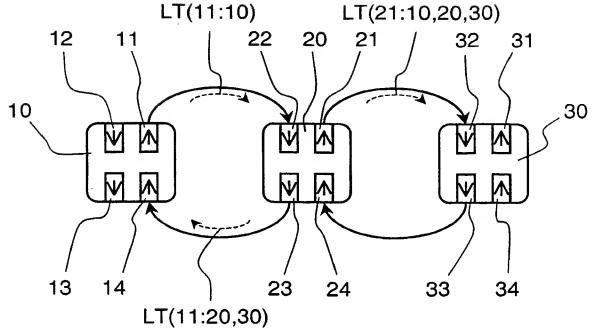



Fig. 7c

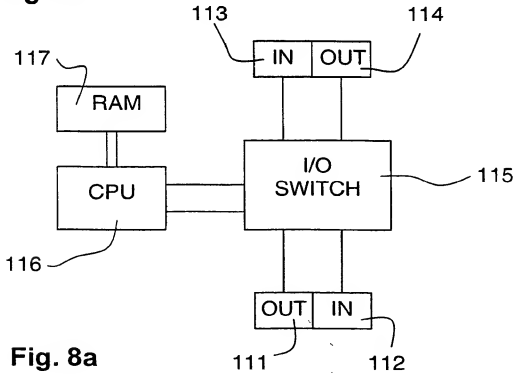



Fig. 8a

| | |
|---|---|
| INTERFACE | 1 |
| LINK ID | 11 |
| REPLY PORT | - |
| FIRST NODE | 10 |
| SECOND NODE | 20 |
| THIRD NODE | 30 |
| ... | ... |
| | |

Fig. 8b

| | |
|---|---|
| INTERFACE | 1 |
| LINK ID | 21 |
| REPLY PORT | 23 |
| FIRST NODE | 10 |
| SECOND NODE | 20 |
| THIRD NODE | 30 |
| ... | ... |
| | |

Fig. 8c

# INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE 99/02169

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: H04L 12/43, H04L 12/52, H04Q 11/04, H04L 12/56
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: H04L, H04Q, H04J

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | GB 2133952 A (INTERNATIONAL STANDARD ELECTRIC CORPORATION), 1 August 1984 (01.08.84), page 1, line 36 - line 44; page 1, line 60 - page 2, line 6; page 2, line 62 - page 3, line 3, figures 1,2, claims 1-19, page 3, line 29 - line 33; page 4, line 19 - line 28<br><br>-- | 1-41 |
| A | US 4287592 A (DANIEL J. PAULISH ET AL), 1 Sept 1981 (01.09.81), column 7, line 29 - line 36; column 8, line 19 - line 57, figures 3,4<br><br>-- | 1-41 |

[X] Further documents are listed in the continuation of Box C. [X] See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" erlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 17 April 2000 | 19 -04- 2000 |
| Name and mailing address of the ISA/<br>Swedish Patent Office<br>Box 5055, S-102 42 STOCKHOLM<br>Facsimile No. +46 8 666 02 86 | Authorized officer<br><br>Erik Johannesson/CL<br>Telephone No. +46 8 782 25 00 |

Form PCT/ISA/210 (second sheet) (July 1992)

# INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE 99/02169

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | US 5732086 A (SONG-CHYAU S. LIANG ET AL), 24 March 1998 (24.03.98), column 1, line 4 - line 30; column 2, line 51 - column 3, line 14, claims 1-10 | 1-41 |
| A | US 4672373 A (KINJI MORI ET AL), 9 June 1987 (09.06.87), claims 1-11 | 1-41 |
| A | US 5440540 A (WILHELM KREMER), 8 August 1995 (08.08.95), claims 1-36, abstract | 1-41 |
| A | WO 9727718 A1 (NEWBRIDGE NETWORKS CORPORATION), 31 July 1997 (31.07.97), see whole document | 1-41 |
| A | WO 9713344 A1 (TELEFONAKTIEBOLAGET LM ERICSSON), 10 April 1997 (10.04.97), see whole document | 1-41 |

# INTERNATIONAL SEARCH REPORT

Information on patent family members

02/12/99

International application No.
PCT/SE 99/02169

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| GB 2133952 A | 01/08/84 | AU 566427 B | 22/10/87 |
| | | AU 2195783 A | 28/06/84 |
| | | BE 895438 A | 22/06/83 |
| | | US 4553233 A | 12/11/85 |
| US 4287592 A | 01/09/81 | NONE | |
| US 5732086 A | 24/03/98 | NONE | |
| US 4672373 A | 09/06/87 | CA 1237803 A | 07/06/88 |
| | | DE 3486204 D,T | 23/12/93 |
| | | DE 3486456 D,T | 30/04/98 |
| | | EP 0147789 A,B | 10/07/85 |
| | | EP 0502555 A,B | 09/09/92 |
| | | JP 2036218 C | 28/03/96 |
| | | JP 7067112 B | 19/07/95 |
| | | JP 60134647 A | 17/07/85 |
| US 5440540 A | 08/08/95 | CA 2065463 A,C | 27/09/93 |
| WO 9727718 A1 | 31/07/97 | AU 1433697 A | 20/08/97 |
| | | AU 6466296 A | 10/02/97 |
| | | CA 2244073 A | 31/07/97 |
| | | EP 0837835 A | 29/04/98 |
| | | EP 0875123 A | 04/11/98 |
| | | GB 9601692 D | 00/00/00 |
| | | JP 11508869 T | 03/08/99 |
| WO 9713344 A1 | 10/04/97 | AU 7079796 A | 28/04/97 |
| | | AU PN573795 D | 00/00/00 |
| | | CA 2231380 A | 10/04/97 |
| | | EP 0853850 A | 22/07/98 |